

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for providing continuous data protection, the method comprising the steps of:

 duplicating a sequence of writes made to a primary volume to a secondary volume, the sequence of writes to the secondary volume being in a sequential order based on a time of a write to the primary volume;

 mapping the writes between the primary volume and the secondary volume; and

 organizing the mapping of the writes into mapping data structures wherein the structures enable the primary volume to be rewound to any point in time.

2. (Canceled)

3. (Currently amended) The method of claim 1 wherein a mapping data structure is configured to keep track of changes to the primary volume between two points in time.

4. (Currently amended) The method of claim 1 wherein a mapping data structure is used to selectively play back portions of ~~a log of each write made to~~ the secondary volume.

5. (Original) The method of claim 1 further comprising the step of merging mapping data structures together to create a new volume that is identical to the primary volume at a previous point in time.

6. (Original) The method of claim 1 further comprising the step of merging mapping data structures that have the same expiration policy to enhance system performance.

7. (Original) The method of claim 1 further comprising the step of fading out data by merging the mapping data structures thereby freeing up blocks that are no longer used.

8. - 11. (Canceled)

12. (Currently amended) A system for providing continuous data protection, the system comprising:

a host computer;

a primary volume for storing a sequence of data written by the host computer;

a secondary volume wherein the sequence of writes made to the primary volume are sequentially duplicated ~~onto~~ on the secondary volume, the sequence writes to the secondary volume being in a sequential order based on a time of a write to the primary volume; and

a continuous data protection system configured to manage the duplication of writes to the secondary volume and to map data between the primary volume and the secondary volume using delta maps mapping data structures.

13. (Cancelled)

14. (Previously presented) The system of claim 12 further comprising a volume manager for mirroring writes to the primary volume and the secondary volume.

15. (Original) The system of claim 14 wherein the volume manager is a software module running on an intelligent storage switch.

16. (Original) The system of claim 14 wherein the volume manager is a software module running on a server.

17. (New) The method of claim 1, wherein the mapping data structures are delta maps.

18. (New) The method of claim 1, wherein the sequential order on the secondary volume is sequential in time.

19. (New) The method of claim 1, wherein the sequential order on the secondary volume is sequential with respect to a storage location on the secondary volume.

20. (New) The system of claim 12, wherein the sequential order on said secondary volume is sequential in time.

21. (New) The system of claim 12, wherein the sequential order on said secondary volume is sequential with respect to a storage location on said secondary volume.

Applicant: Stager et al.
Application No.: 10/771,613

22. (New) The system of claim 12, wherein a mapping data structure is configured to keep track of changes to said primary volume between two points in time.

23. (New) The system of claim 12, wherein a mapping data structure is used to selectively play back portions of said secondary volume.

24. (New) The system of claim 12, wherein said continuous data protection system is further configured to merge mapping data structures together to create a new volume that is identical to said primary volume at a previous point in time.

25. (New) The system of claim 12, wherein said continuous data protection system is further configured to merge mapping data structures that have the same expiration policy to enhance system performance.

26. (New) The system of claim 12, wherein said continuous data protection system is further configured to fade out data by merging the mapping data structures, thereby freeing up blocks that are no longer used.

27. (New) The system of claim 12, wherein the mapping data structures are delta maps.

28. (New) A method for continuously protecting data on a primary volume, comprising the steps of:

creating an initial snapshot of the primary volume, the initial snapshot located on a secondary volume;

duplicating a sequence of writes made to the primary volume to a write log on the secondary volume, the sequence of writes to the write log being in a sequential order based on a time of a write to the primary volume;

determining whether a snapshot has been triggered;

inserting a marker into the write log if a snapshot has been triggered; and

creating a mapping data structure from the write log, wherein the mapping data structure represents changes to the primary volume between two points in time, whereby the method continuously protects the primary volume from the time of the initial snapshot.

29. (New) The method of claim 28, wherein the sequential order on the secondary volume is sequential in time.

30. (New) The method of claim 28, wherein the sequential order on the secondary volume is sequential with respect to a storage location on the secondary volume.

31. (New) The method of claim 28, wherein the creating step includes converting the write log to a block-ordered mapping data structure.

32. (New) The method of claim 28, wherein the mapping data structure is used to play back portions of the write log.

33. (New) The method of claim 28, further comprising the step of: periodically merging mapping data structures from a time of a previous snapshot to the current time, whereby the merging step optimizes the mapping data structures.

34. (New) The method of claim 33, wherein the merging step is performed according to a predetermined policy.

35. (New) The method of claim 33, wherein the merging step is performed after a predetermined number of writes to the primary volume.

36. (New) The method of claim 33, wherein the merging step is performed after a predetermined period of time.

37. (New) The method according to claim 33, wherein the merging step includes merging mapping data structures having the same retention policy.

38. (New) A computer-readable storage medium containing a set of instructions for a general purpose computer, the set of instructions comprising:

a duplicating code segment for duplicating a sequence of writes made to a primary volume to a secondary volume, the sequence of writes to the secondary volume being in a sequential order based on a time of a write to the primary volume;

a mapping code segment for mapping the writes between the primary volume and the secondary volume; and

an organizing code segment for organizing the mapping of the writes into mapping data structures wherein the structures enable the primary volume to be rewound to any point in time.

39. (New) A computer-readable storage medium containing a set of instructions for a general purpose computer, the set of instructions comprising:

a first creating code segment for creating an initial snapshot of a primary volume, the initial snapshot located on a secondary volume;

a duplicating code segment for duplicating a sequence of writes made to the primary volume to a write log on the secondary volume, the sequence of writes to the write log being in a sequential order based on a time of a write to the primary volume;

a determining code segment for determining whether a snapshot has been triggered;

an inserting code segment for inserting a marker into the write log if a snapshot has been triggered; and

a second creating code segment for creating a mapping data structure from the write log, wherein the mapping data structure represents changes to the primary volume between two points in time, whereby the primary volume is continuously protected from the time of the initial snapshot.